RECEIVED
CENTRAL FAX CENTER
SEP 0 5 2006

CLAIM AMENDMENTS

The following is a complete listing of the pending claims:

1. (currently amended) A programmable logic device, comprising:

a configuration memory operable to store configuration data, the configuration memory including logic blocks each associated with a respective status indicator indicating whether the associated logic block is configured as random access memory;

a checksum calculation engine operable to cyclically process the configuration data during operation of the programmable logic device using an error detection algorithm, the checksum calculation engine calculating a checksum during each ~~calculation~~ cycle, the checksum calculation excluding configuration data for logic blocks whose respective status indicator indicates the logic block is configured as random access memory; and

a checksum comparator configured to compare the checksum calculated by the checksum calculation engine in a given ~~calculation~~ cycle with a previously-calculated checksum so as to verify the integrity of the configuration data.

2. (previously presented) The programmable logic device of claim 1, wherein the checksum calculation engine is a CRC calculation engine, and wherein the checksum comparator is a CRC checksum comparator.

3. (previously presented) The programmable logic device of claim 1, wherein the checksum calculation engine is a parity bit calculation engine, and the checksum comparator is a parity bit checksum comparator.

4. (cancelled)

5. (previously presented) The programmable logic device of claim 2, wherein the CRC calculation engine comprises a linear feedback shift register (LFSR).

6. (previously presented) The programmable logic device of claim 2, further comprising:

a register operable to store a predetermined CRC checksum, wherein the previously-calculated checksum used by the CRC checksum comparator is the predetermined CRC checksum.

7. (previously presented) The programmable logic device of claim 1, wherein the checksum calculation engine is configured to calculate an initial checksum during its initial cycle, and wherein the previously-calculated checksum used by the checksum comparator is the initial checksum.

8. (cancelled).

9. (previously presented) The programmable logic device of claim 1, further comprising:

a configurable logic core, wherein the checksum calculation engine and the checksum comparator are each implemented by configuring the logic core.

10. (previously presented) The programmable logic device of claim 1, wherein the checksum calculation engine and the checksum comparator each comprises dedicated hardware.

11. (currently amended) A programmable logic device, comprising:

a configuration memory operable to store configuration data, the configuration data including logic blocks each associated with a respective status indicator indicating whether the associated logic block is configured as random access memory; and

error-checking means for cyclically processing the configuration data during operation of the programmable logic device with an error detection algorithm so as to calculate a checksum during each calculation cycle and for comparing the checksum calculated in a given calculation cycle with a previously-calculated checksum, wherein the error-checking means is configured to exclude configuration data for a logic block whose respective status indicator indicates the logic block is configured as random access memory.

12. (previously presented) The programmable logic device of claim 11, wherein the error-checking means is configured to cyclically process the configuration data during operation of the programmable logic device with a CRC algorithm so as to calculate a CRC checksum during each cycle and to compare the CRC checksum calculated in a given cycle with a previously-calculated CRC checksum.

13. (previously presented) The programmable logic device of claim 12, wherein error-checking means is configured to compare the CRC checksum calculated in a given cycle with a predetermined CRC checksum.

14. (cancelled)

15. (currently amended) A method, comprising:

(a) configuring a programmable logic device with configuration data, the configuration data including status indicators for associated logic blocks to indicate whether the associated logic block is configured as random access memory;

operating the configured programmable logic device;

(b) during operation of the programmable logic device, cyclically processing the configuration data using an error-detection algorithm to generate a checksum during each calculation cycle, wherein the cyclical processing of the configuration data excludes configuration data for logic blocks whose status indicators indicate are configured as random access memory; and

(c) after each calculation cycle, comparing the generated checksum with a previously-calculated checksum to verify the integrity of the configuration data.

16. (previously presented) The method of claim 1, wherein act (b) comprises cyclically processing the configuration data using a CRC algorithm to generate a CRC checksum during each cycle, and wherein act (c) comprises comparing the generated CRC checksum with a previously-calculated CRC checksum.

17. (previously presented) The method of claim 15, wherein act (a) comprises configuring the programmable logic device with configuration data such that the programmable logic device is configured to perform acts (b) and (c).

18. (cancelled)

19. (previously presented) The method of claim 15, further comprising:

asserting a good configuration memory flag if the comparison in act (c) indicates that the integrity of the configuration memory has not been corrupted.

20. (previously presented) The method of claim 19, further comprising:

if the good configuration memory flag is not asserted, reconfiguring the programmable logic device with the configuration data.

Claims 21 – 24. (cancelled).

25. (new) The method of claim 20, wherein the programmable logic device includes a non-volatile memory, and wherein reconfiguring the programmable logic device with the configuration data comprises retrieving the configuration data from non-volatile memory.

26. (new) The programmable logic device of claim 1, wherein the status indicator comprises a status bit.

27. (new) The programmable logic device of claim 1, wherein the status indicator comprises the address of a logic block to be excluded, the address stored with the configuration data in configuration memory.

28. (new) A programmable logic device, comprising:

a configuration memory operable to store configuration data, the configuration memory including logic blocks and memory blocks;

a checksum calculation engine operable to cyclically process the configuration data during operation of the programmable logic device using an error detection algorithm, the checksum calculation engine calculating a checksum during each cycle, the checksum calculation excluding a logic block or a memory block whose address is stored with the configuration data in the configuration memory; and

a checksum comparator configured to compare the checksum calculated by the checksum calculation engine in a given cycle with a previously-calculated checksum so as to verify the integrity of the configuration data.